International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

UPI Sentinel: A Transaction-Contextual Deep Learning Framework for UPI Fraud Detection

Ridham Taneja, Rahul Kumar Patel, Navneet Kumar, Paritosh Mukherjee, Arpita Singh
Department of Computer Science and Engineering (AI & ML)
Raj Kumar Goel Institute of Technology
Ghaziabad, India

***Abstract*—The exponential adoption of the Unified Payments Interface (UPI) has revolutionized digital finance, yet it has concurrently created a fertile ground for sophisticated financial fraud. Although existing fraud detection systems employing traditional Machine Learning (ML) classifiers—such as Random Forests and Support Vector Machines—have shown efficacy in identifying static anomalies, they often struggle to capture the complex, sequential dependencies inherent in modern transactional fraud (e.g., high-frequency layering or mule account chaining). To address this limitation, this paper proposes a novel Hybrid Deep Learning framework specifically adapted for the UPI domain. Our architecture synergizes Convolutional Neural Networks (CNN) to extract granular spatial features from transaction attributes with Long Short-Term Memory (LSTM) networks to model temporal dependencies across transaction sequences. Furthermore, we integrate an Attention Mechanism to dynamically weight the most critical time steps in a transaction history, allowing the model to focus on subtle indicators of fraud that are often diluted in long sequences. The framework is evaluated on the PaySim dataset, selected for its extreme class imbalance and similarity to real-world UPI transaction topologies. Experimental results demonstrate strong discriminative performance, achieving an ROC-AUC of 0.997 and a PR-AUC of 0.872. At an optimized decision threshold, the model attains a fraud detection F1-score of 0.85, with a precision of 0.91 and recall of 0.80, while maintaining a very low false positive rate. These findings indicate that combining spatio-temporal modeling with attention-based aggregation provides a robust and adaptable solution for fraud risk scoring in large-scale digital payment ecosystems.**

***Index Terms*—UPI Fraud Detection, Hybrid Deep Learning, CNN-LSTM, Attention Mechanism, Imbalanced Classification, Transaction Context**

I. Introduction

The proliferation of mobile internet connectivity has positioned the Unified Payments Interface (UPI) as a leading global infrastructure for real-time peer-to-peer (P2P) payments. According to official statistics released by the National Payments Corporation of India (NPCI), UPI processed more than 110 billion transactions in the first half of 2025 alone [1]. Although this rapid adoption has expanded financial accessibility, it has concurrently increased the attack surface for cybercriminals. As transactions become instantaneous, the window available for fraud detection narrows, necessitating accurate near real-time decision-making mechanisms.

Financial fraud has evolved from isolated unauthorized access to coordinated schemes such as Authorized Push Payment (APP) fraud, layering, and mule account networks. Traditional fraud detection systems often rely on rule-based engines or static Machine Learning (ML) classifiers, including Logistic Regression and Random Forests [2]. Although effective at detecting point anomalies (e.g., unusually large transaction amounts), such approaches typically fail to capture short-term temporal dependencies that characterize sophisticated fraud strategies. Fraudsters frequently execute a sequence of small "testing" transactions prior to a major withdrawal—a behavioral pattern that static models often overlook.

To address these limitations, we propose *UPI Sentinel*, a hybrid deep learning framework tailored to the high-throughput UPI ecosystem. Rather than introducing a novel standalone architecture, this study emphasizes the strategic integration of complementary deep learning paradigms to enhance detection performance [3]. Convolutional Neural Networks (CNNs) are employed to extract spatial representations from individual transaction records, while Long Short-Term Memory (LSTM) networks model evolving behavioral sequences. An attention mechanism is further incorporated to refine temporal aggregation [4]. By dynamically assigning greater importance to high-risk time steps—such as sudden bursts of transactional activity—the model ensures that classification decisions are guided by the most informative behavioral signals.

The primary contributions of this study are summarized as follows:

- **Hybrid Spatio-Temporal Architecture:** Integration of a domain-tuned CRNN framework leveraging 1D-CNN layers for spatial feature extraction and BiLSTM layers for temporal modeling tailored to UPI transaction sequences.
- **Context-Aware Data Pipeline:** Implementation of a global sliding window strategy that converts isolated transaction logs into short-term sequential representations for contextual risk assessment.
- **Attention-Based Temporal Aggregation:** Incorporation of an attention mechanism to dynamically weight informative transaction steps, isolating high-risk patterns while suppressing benign transactional noise.
- **Probabilistic Risk Scoring:** Utilization of a sigmoid-based continuous risk scoring mechanism enabling flexible decision thresholds for deployment scenarios such as transaction blocking or step-up authentication.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

- **Commercial Viability Analysis:** Comprehensive evaluation on the highly imbalanced PaySim dataset with post-training threshold calibration, demonstrating practical applicability by minimizing false positives while maintaining strong fraud detection sensitivity in real-world deployment settings.

II. Related Work

A. Evolution from Static Classifiers to Deep Learning

Financial fraud detection initially relied on Machine Learning (ML) classifiers such as Logistic Regression, Support Vector Machines (SVM)and Random Forests [2], [5]. Although computationally efficient and interpretable, these models analyze transactions independently and are therefore effective at identifying isolated anomalies (e.g., extreme monetary values). However, they struggle to capture coordinated fraud behaviors that unfold across short transaction sequences. This challenge is further intensified by the severe class imbalance typical of financial datasets, where minority fraud cases are easily overshadowed by legitimate activity [6].

To address these limitations, deep learning approaches were introduced to enhance representation learning from transactional data [7]. Fully connected Deep Neural Networks (DNNs) demonstrated improved modeling of non-linear feature interactions compared to traditional ML methods. Nevertheless, feedforward architectures remain inherently static and do not explicitly model the temporal evolution of user behavior. This limitation motivated the adoption of sequential modeling techniques capable of capturing behavioral progression across consecutive transactions.

B. Sequential Modeling and Attention Mechanisms

Sequential deep learning architectures treat fraud detection as a temporal modeling task. Recurrent Neural Networks (RNNs) marked an early step in this direction but exhibited limitations in preserving long-range contextual dependencies. Long Short-Term Memory (LSTM) networks were subsequently adopted to better retain relevant historical information across transaction streams [8]. Bidirectional LSTMs (BiLSTMs) further extend this capability by incorporating contextual information from both past and future positions within a transaction window [9].

Although recurrent architectures improve temporal representation, they typically aggregate sequence information uniformly, which may dilute critical fraud indicators within noisy histories. Attention mechanisms mitigate this issue by assigning adaptive importance to informative time steps, allowing models to emphasize high-risk behavioral segments [10]. Hybrid frameworks combining convolutional layers for localized feature extraction with BiLSTM and attention components have demonstrated promising results in sequential learning applications [11].

Despite these advancements, existing approaches frequently rely on rigid user-level grouping or long transaction histories, which may be impractical in privacy-sensitive or high-volume environments. Moreover, limited work jointly addresses short-term sequential modeling and extreme class imbalance without explicit user profiling. This study addresses these gaps by introducing a transaction-contextual framework that integrates CRNN architectures with attention-based aggregation under a global sliding window formulation.

III. Methodology

A. Dataset Description

This study utilizes PaySim, a synthetic mobile money dataset introduced by Lopez-Rojas et al. [12]. The dataset comprises 6,362,620 transactions, of which 8,213 (0.13%) are labeled as fraudulent. Each record includes key attributes such as transaction amount, transaction type (e.g., Transfer, Cash-Out), temporal step index, and pre- and post-transaction balances for both sender and receiver accounts. The binary label *isFraud* indicates whether a transaction is fraudulent or legitimate. Owing to its severe class imbalance and sequential structure, PaySim provides a practical benchmark for evaluating context-aware fraud detection methods under realistic constraints [6].

B. Dataset Preprocessing and Feature Encoding

To prepare the dataset for the model, a set of preprocessing steps are performed to ensure data consistency, numerical compatibility, and stable learning behavior [2]. The categorical transaction type attribute is transformed into numerical form using label encoding, enabling its integration into neural network models while preserving distinctions between different transaction categories [5].

Numerical features, including transaction amount and account balance attributes, are examined for missing values. Any missing entries are replaced with zero, which is appropriate for the PaySim dataset as such values typically correspond to inactive or non-participating accounts rather than corrupted records [12]. To improve numerical stability and accelerate convergence during training, all numerical features are standardized using z-score normalization [13]

C. Transaction Sequence Construction

To incorporate short-term transactional context into the fraud detection process, transactions are transformed into fixed-length sequences using a global sliding window protocol.

Prior work has explored sequential transaction modeling using sliding window mechanisms to capture short-term temporal dependencies [14], [15]. Unlike user-grouped sequence construction, which requires persistent user identifiers and sufficient transaction history per account, a global sliding window enables contextual modeling over the complete transaction stream without discarding data or introducing user-level dependencies.

Rather than treating each transaction independently, this approach enables the model to observe a small temporal neighborhood of transactions and capture contextual patterns

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

occurring over consecutive transaction steps. First, all transactions are ordered according to their temporal step to preserve temporal ordering within each constructed sequence.

This process is illustrated in Fig.1, where overlapping windows generate sequential samples, and each sequence is labeled using the final transaction in the window.

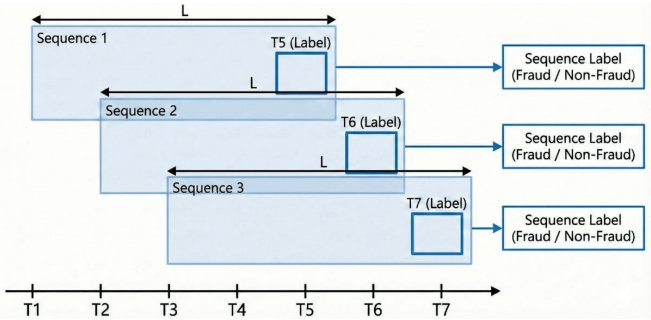

Fig. 1. Global sliding window protocol for transaction sequence construction.

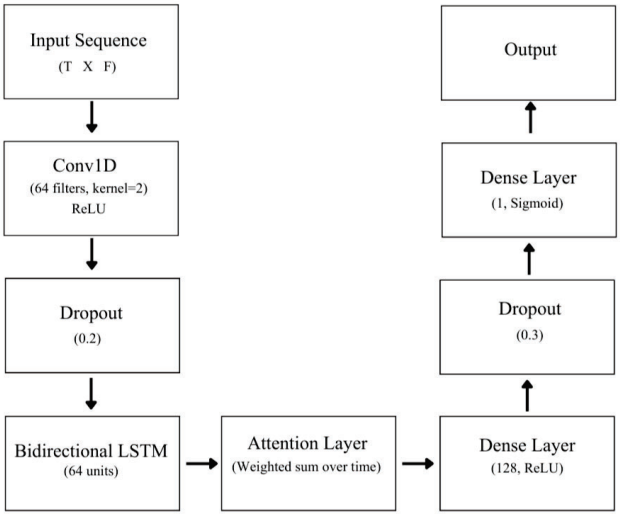

Fig. 2. Proposed CRNN architecture for transaction-level fraud detection

D. Class Imbalance Handling Using SMOTE

Financial transaction datasets are inherently imbalanced, with fraudulent cases representing only a small fraction of total transactions. Such imbalance can bias models toward the majority class, reducing fraud detection sensitivity.

The dataset is partitioned into training and testing sets using an 80:20 split while preserving the original class distribution. To prevent information leakage, the Synthetic Minority Oversampling Technique (SMOTE) is applied exclusively to the training data [16], while the test set remains unchanged.

SMOTE generates synthetic minority samples through interpolation between existing fraud instances. In this study, oversampling is performed after sequence construction to preserve the temporal structure of transaction windows. This strategy improves minority-class representation during training while ensuring unbiased evaluation on the original distribution.

E. Proposed CRNN Architecture

To model short-term transactional patterns within constructed transaction sequences, this study adopts a Convolutional Recurrent Neural Network (CRNN) architecture augmented with an attention mechanism. The architecture combines convolutional layers for local feature extraction, recurrent layers for temporal dependency modeling, and an attention module to emphasize informative transaction steps. This integrated design enables effective learning from sequential transaction data while maintaining robustness to noise and variability commonly present in financial transactions. The architecture follows the general CRNN paradigm that combines convolutional and recurrent layers for sequential data modeling [11], and incorporates an attention mechanism to emphasize informative temporal patterns [14]. The proposed architecture is illustrated in Fig. 2.

1) Convolutional Feature Extraction: The architecture begins with a one-dimensional convolutional layer designed to extract local patterns from transaction sequences. Given an input sequence of transactions, the convolutional layer operates along the temporal dimension, enabling the model to learn short-range dependencies and interactions between consecutive transactions. Specifically, the Conv1D layer applies 64 filters with a kernel size of 2, allowing the model to capture fine-grained interactions between adjacent transaction steps [7]. A Rectified Linear Unit (ReLU) activation function is used to introduce non-linearity and enhance the model's ability to learn complex transactional patterns [17].

To improve generalization and mitigate overfitting, a dropout layer with a rate of 0.2 is applied immediately after the convolutional operation [18]. The resulting feature maps preserve the sequential structure of the input while providing enriched representations that are subsequently passed to the recurrent component for temporal modeling.

2) Bidirectional LSTM for Temporal Modeling: Although convolutional layers capture local transactional patterns, modeling longer-range temporal dependencies requires a recurrent mechanism. To achieve this, the architecture employs a Bidirectional Long Short-Term Memory (BiLSTM) layer that processes the convolutional feature sequences in both forward and backward temporal directions [9]. This bidirectional structure enables the model to capture dependencies that may occur both before and after a given transaction within a sequence.

The BiLSTM layer consists of 64 hidden units in each direction and is configured to return the full sequence of hidden states. This allows subsequent layers to access temporal information at each transaction step rather than a single aggregated representation. LSTM networks are particularly well-suited for sequential financial data due to their ability to mitigate vanishing gradient issues and retain relevant information over time [19].

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

By incorporating bidirectional processing, the model gains a more comprehensive understanding of transaction dynamics within each sequence, which is especially beneficial in fraud detection scenarios where contextual cues may span multiple neighboring transactions [8]. The sequence-aware representations produced by the BiLSTM layer are then forwarded to the attention mechanism for selective temporal aggregation.

3) Attention Mechanism: Although recurrent layers model temporal dependencies across transaction sequences, not all transactions within a sequence contribute equally to fraud detection. To address this, an attention mechanism is integrated after the BiLSTM layer to enable selective emphasis on informative transaction steps. The attention layer operates on the sequence of hidden states produced by the BiLSTM and learns a set of importance weights over time. These weights indicate the relative contribution of each transaction in the sequence toward the final prediction. A weighted aggregation of the hidden states is then performed to produce a fixed-length context vector, which summarizes the most relevant temporal information for classification.

By dynamically focusing on critical transactions—such as sudden balance changes or unusually large transfers—the attention mechanism enhances interpretability and allows the model to prioritize salient temporal patterns. This approach has been shown to improve performance in sequence modeling tasks by enabling adaptive feature weighting rather than uniform temporal aggregation [14], [10].

4) Output Layer and Prediction: The context vector produced by the attention mechanism is passed through a fully connected layer with 128 neurons and a ReLU activation function to learn higher-level abstractions from the aggregated temporal representation. To further enhance generalization and reduce overfitting, a dropout layer with a rate of 0.3 is applied before the final classification stage.

The output layer consists of a single neuron with a sigmoid activation function, which maps the learned representation to a probability score indicating the likelihood of a transaction being fraudulent. This probabilistic output enables flexible decision-making through threshold adjustment, allowing the model to balance precision and recall according to deployment requirements [20].

IV. Experiments and Results

A. Experimental Setup

The proposed model was implemented using Python 3 with the TensorFlow/Keras deep learning framework and the Imbalanced-Learn library for class imbalance handling. All experiments were conducted in a Google Colab environment utilizing an NVIDIA Tesla T4 GPU for accelerated training.

1) Dataset and Partitioning: The PaySim dataset [12], after preprocessing and transformation into sequential data using the global sliding window protocol, was used for all experiments. The dataset was divided using a stratified split, allocating 80% of the data for training and 20% for testing, ensuring that the original class distribution was preserved in both sets.

To address the severe class imbalance inherent in fraud detection, the Synthetic Minority Over-sampling Technique (SMOTE) was applied *only* to the training data after sequence construction, thereby preventing data leakage [16]. The test set was kept unchanged and used solely for evaluation.

To ensure computational efficiency while maintaining statistical relevance, model training was performed on a stratified and balanced subset of 100,000 samples drawn from the resampled training data.

2) Hyperparameter Configuration: The model was trained using the Adam optimizer with a learning rate of 0.001, and binary cross-entropy was employed as the loss function. In addition to SMOTE, class-weighted training was utilized to further mitigate residual class imbalance during model optimization. The key hyperparameters used in the experiments are summarized in Table I.

TABLE I
Hyperparameter Configuration

Parameter	Value
Optimizer	Adam (Learning Rate = 0.001)
Loss Function	Binary Cross-Entropy
Batch Size	64
Maximum Epochs	10
Early Stopping	Patience = 3 (Validation Loss)
Conv1D Filters	64 (Kernel Size = 2)
BiLSTM Units	64
Dropout Rates	0.2 (after CNN), 0.3 (after Dense)

B. Evaluation Metrics and Threshold Tuning

The model achieved an overall accuracy of 99.89% on the test set. Although this figure appears exceptionally high, it was deemed insufficient and misleading for performance evaluation due to the extreme class imbalance (0.13% fraud prevalence). In a dataset where 99.87% of transactions are legitimate, a trivial model that predicts "Normal" for every single transaction would still achieve 99.87% accuracy while failing to detect a single fraud case.

Consequently, accuracy was discarded as a primary metric. Instead, performance was evaluated using F1-Score , Recall, Precision, and the Area Under the Curve (AUC) for both ROC and Precision-Recall (PR) characteristics [6].

To minimize false positives while maintaining high fraud detection capability, we applied threshold tuning. The decision threshold was shifted from the default 0.5 to 0.9975, based on the maximization of the F1-score [20]. This adjustment resulted in a model that is highly conservative in flagging transactions, thereby preserving the user experience for legitimate customers.

C. Confusion Matrix Analysis

The classification performance at the optimal threshold ($\tau = 0.9975$) is visualized in Fig. 4. The confusion matrix reveals that the model successfully identified 1,253 fraudulent transactions (True Positives) out of a total of 1,643 cases.

Notably, the model demonstrated exceptional specificity, misclassifying only 68 legitimate transactions as fraud (False

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

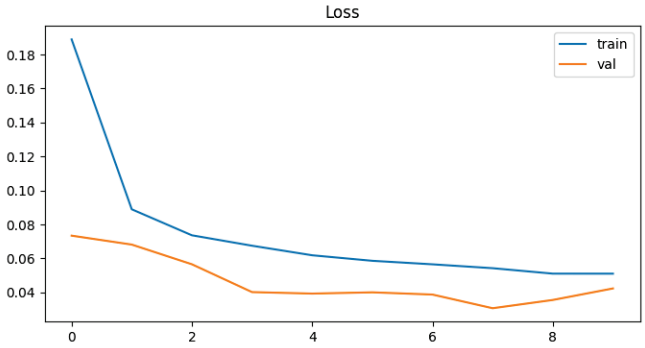

Fig. 3. Training and validation loss across epochs for the proposed hybrid model..

Positives) out of over 1.27 million legitimate samples. This low False Positive rate is critical in financial fraud detection systems, as excessive false alarms can lead to customer dissatisfaction and operational bottlenecks. The low number of False Negatives (390) indicates that while the model is highly precise, it still captures the vast majority of fraudulent activity.

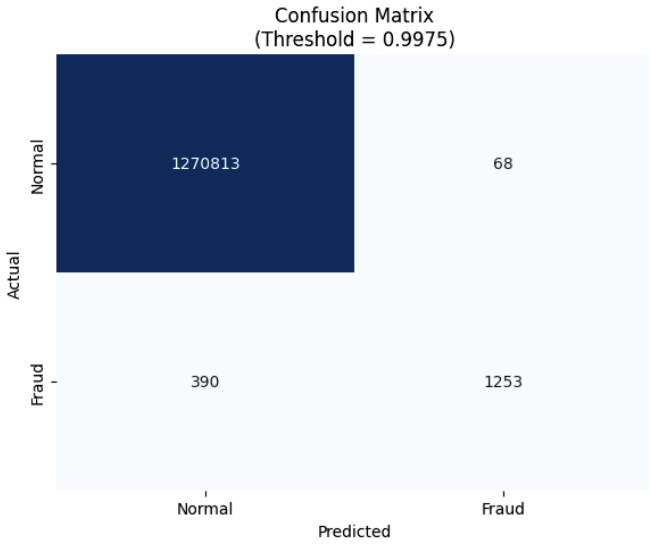

Fig. 4. Confusion matrix at threshold $\tau = 0.9975$, illustrating the classification outcomes of the proposed hybrid model.

D. Receiver Operating Characteristic (ROC) Curve

The model's ability to discriminate between fraudulent and non-fraudulent classes is demonstrated in Fig. 5. The Receiver Operating Characteristic (ROC) curve exhibits a steep initial ascent, indicating a high True Positive Rate (Sensitivity) even at very low False Positive Rates.

The Area Under the Curve (ROC-AUC) was calculated to be 0.9972. This near-perfect score confirms that the classifier maintains robust separability between classes across various decision thresholds, validating the efficacy of the selected feature set and model architecture.

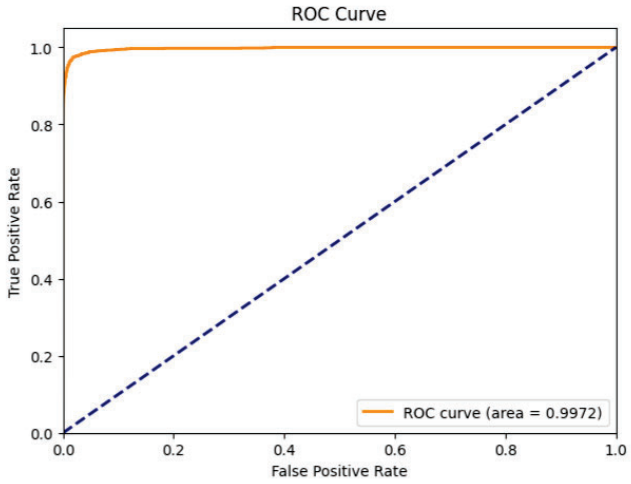

Fig. 5. Receiver Operating Characteristic (ROC) curve with AUC = 0.9972 for the proposed hybrid model.

E. Precision-Recall (PR) Curve Analysis

Given the class imbalance, the Precision-Recall (PR) curve serves as a more reliable performance indicator than the ROC curve. As shown in Fig. 6, the model maintains high precision across a substantial range of recall values.

The Area Under the Precision-Recall Curve (PR-AUC) achieved was 0.8718. Unlike the ROC metric, which can be overly optimistic in imbalanced scenarios, the PR-AUC specifically highlights the model's success in the minority class (Fraud). A score of 0.87 represents a strong trade-off, ensuring that when the model predicts fraud, it is highly likely to be correct, without missing a significant portion of actual fraud cases.A consolidated summary of the model's performance at the optimized threshold is presented in Table II.

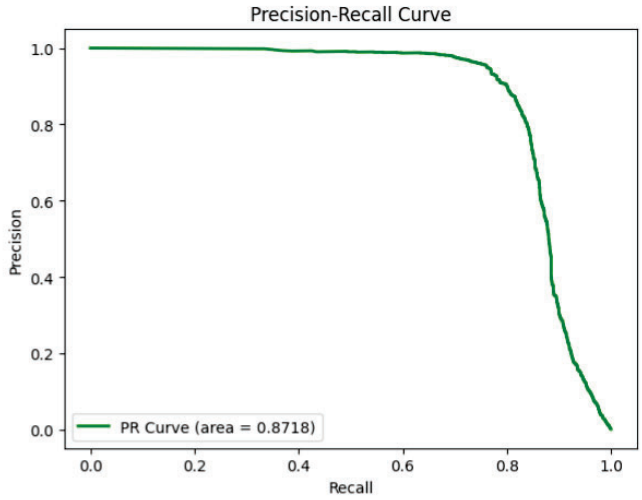

Fig. 6. Precision-Recall curve with PR-AUC = 0.8718 for the proposed hybrid model.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

TABLE II
SUMMARY OF MODEL PERFORMANCE

Metric	Value
Optimal Threshold (τ)	0.9975
ROC-AUC	0.9972
PR-AUC	0.8718
False Positives (FP)	68
True Positives (TP)	1,253
False Negatives (FN)	390

V. CONCLUSION AND FUTURE WORK

This study demonstrates the effective application of a hybrid deep learning framework for transaction-contextual fraud detection using the PaySim mobile money simulation dataset. By integrating convolutional, bidirectional recurrent, and attention-based components within a unified pipeline, the proposed approach captures short-term transactional dependencies while maintaining robustness under severe class imbalance.

To address the low fraud prevalence (0.13%), the Synthetic Minority Over-sampling Technique (SMOTE) was applied exclusively during training to enhance minority-class representation without introducing evaluation bias. Combined with post-training threshold optimization ($\tau = 0.9975$), this strategy enables the model to balance fraud sensitivity and false positive control effectively. The final system achieved an ROC-AUC of 0.9972 and a False Positive Rate of 0.005%, demonstrating strong discriminative capability in highly imbalanced financial settings.

While PaySim provides a realistic approximation of mobile money transactions, certain attributes (e.g., post-transaction balances) may reflect information not immediately available at real-time decision points. Nevertheless, the proposed framework does not depend on any single feature; instead, it learns contextual patterns across multiple transactional attributes and short temporal windows. This design supports robust prediction even when individual features are delayed, partially observable, or noisy.

While the current framework demonstrates strong performance, several avenues exist for further enhancement:

- Integration of Transformer-based architectures to enhance long-range dependency modeling.
- Deployment-oriented latency evaluation in real-time streaming environments.
- Incorporation of explainability techniques (e.g., SHAP) to improve interpretability and regulatory transparency.

In practical deployments, access to richer contextual signals—such as device-level attributes, behavioral velocity indicators, and network-level risk features—may further enhance predictive performance. Integrating such signals alongside complementary techniques (e.g., graph-based risk modeling) represents a promising direction for improving scalability and adaptability in large-scale financial systems.

REFERENCES

[1] National Payments Corporation of India, "UPI product statistics," 2025, accessed: 2025-01-23. [Online]. Available: https://www.npci.org.in/what-we-do/upi/product-statistics
[2] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Waltham, MA, USA: Morgan Kaufmann, 2011.
[3] S. Jagadeesan, K. S. Arjun, G. Dhanika, G. Karthikeyan, and K. Deepika, "UPI fraud detection using machine learning," in *Challenges in Information, Communication and Computing Technology*, 1st ed., V. Sharmila *et al.*, Eds. London, U.K.: CRC Press, 2024, pp. 755–760.
[4] I. Akour, N. Mohamed, and S. Salloum, "Hybrid CNN-LSTM with attention mechanism for robust credit card fraud detection," *IEEE Access*, vol. 13, pp. 1–12, 2025.
[5] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2019.
[6] A. Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," in *Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI)*, Cape Town, South Africa, Dec. 2015, pp. 159–166.
[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
[8] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000.
[9] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.
[10] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, San Diego, CA, USA, Jun. 2016, pp. 1480–1489.
[11] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2298–2304, Nov. 2017.
[12] E. A. Lopez-Rojas, A. Elmir, and S. Axelsson, "PaySim: A financial mobile money simulator for fraud detection," in *Proceedings of the 28th European Modeling and Simulation Symposium (EMSS)*, Larnaca, Cyprus, 2016.
[13] S. Patro and K. K. Sahu, "Normalization: A preprocessing stage," *arXiv preprint arXiv:1503.06462*, 2015. [Online]. Available: https://arxiv.org/abs/1503.06462
[14] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, 2015.
[15] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long- and short-term temporal patterns with deep neural networks," in *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR)*, Ann Arbor, MI, USA, 2018, pp. 95–104.
[16] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002.
[17] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML)*, Haifa, Israel, Jun. 2010, pp. 807–814.
[18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, Jun. 2014.
[19] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, Mar. 1994.
[20] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS ONE*, vol. 10, no. 3, p. e0118432, Mar. 2015.